Automated Conflict Detection Between Medical Care Pathways

Weber, Philip; Ferreira Filho, Joao Bosco; Bordbar, Behzad; Lee, Mark; Litchfield, Ian; Backman, Ruth

DOI:
10.1002/smr.1898

Document Version
Peer reviewed version

Citation for published version (Harvard):
Weber, P, Ferreira Filho, JB, Bordbar, B, Lee, M, Litchfield, I & Backman, R 2018, 'Automated Conflict Detection Between Medical Care Pathways', *Journal of software: Evolution and Process*, vol. 30, no. 7, e1898. https://doi.org/10.1002/smr.1898

JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS
J. Softw. Evol. and Proc. 2017; **00**:1–23
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/smr

Automated Conflict Detection Between Medical Care Pathways

P. Weber[1*], J. B. F. Filho[1], B. Bordbar[1], M. Lee[1], I. Litchfield[2], R. Backman[2]

[1]*School of Computer Science, University of Birmingham, B15 2TT, UK*
[2]*Institute of Applied Health Research, University of Birmingham, B15 2TT, UK*

SUMMARY

Clinical guidelines specify sequences of steps (care pathways) to treat patients with single conditions. Increasingly many patients exhibit 'multimorbidity', several chronic conditions needing concurrent treatment. However, applying multiple guidelines in parallel can lead to conflicts, e.g. between prescribed drugs, lifestyle intervention recommendations, or treatment schedules. In computer science, process languages used to design and reason about software development and business process management are similar to clinical pathways. Using formal model transformation, composition and analysis methods, models can be combined and conflicts detected and resolved. We propose BPMN+V, a data-driven formal model for clinical care pathways, as an extension of BPMN. We describe a method for conflict detection using a transformation of BPMN+V to Coloured Petri Nets, and a state-space method for detection of conflict in composed models. We present results from a case study, showing that common conflicts are successfully detected, and propose extension to a complete framework for efficiently recommending resolutions to medical conflicts in composed care pathway models. Copyright © 2017 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: BPMN, Workflow models, Coloured Petri Nets, Model Transformation, Clinical Guidelines, Care Pathways, Conflict Detection, Multimorbidity

1. INTRODUCTION

The Business Process Model and Notation [1] (BPMN) is the *de facto* and also ISO[†] standard [2] for process modelling, providing support for modelling control flow, data flow and resource allocation. BPMN's intuitive graphical model [3] is particularly suitable for capturing business processes by domain experts who may not have development skills [4, 5, 6]. The ability to handover BPMN specifications to automatically assist execution via languages such as Business Process Execution Language (BPEL [7]) reduces the time and cost from the design of a business process to its production. As a result, BPMN is widely adopted [8] within industry[‡] and via various open source Business Process Management (BPM) tools. BPMN has been widely used in various application domains [9], including government [10], software development [11] and service management [12], construction [13], education [14], and healthcare (e.g. [4, 5, 6, 15, 16, 17]).

Clinical guidelines document the best available evidence for care of patients with specific medical conditions ('morbidities'). In the United Kingdom (UK) they are used in combination with national guidance and local National Health Service (NHS) policy to provide appropriate care in a local

*Correspondence to: Philip Weber, School of Computer Science, University of Birmingham, B15 2TT, UK. Email: dr.philip.weber@ieee.org

Contract/grant sponsor: EPSRC; contract/grant number: EP/M014401/1
[†]International Organization for Standardization.
[‡]E.g. SAP (https://go.sap.com/), IBM Websphere (https://www.ibm.com/software/websphere/).

context. Guidelines have to date been published by the National Institute of Clinical Excellence [18] for 253 individual morbidities (e.g. Type Two Diabetes Mellitus [19] and Chronic Obstructive Pulmonary Disease (COPD) [20]). Flowchart-style care pathways can be generated from these guidelines and have been found to improve patient outcomes [21]. These are informal in the sense that they are not endowed with a rigorous mathematical semantics.

Patients with multiple concurrent chronic conditions (referred to as 'multimorbidity') are rapidly rising in prevalence in the UK [22]. Guidance relating to the management of these patients is limited [23, 24], since guidelines focus on single morbidities, and when new treatments or interventions are assessed for effectiveness within a clinical trial setting, co-morbidities are excluded where possible to limit confounding the findings. Therefore it is challenging to form guidance tailored to a patient as an individual, rather than focussed on each clinical condition in isolation [25, 26]. In this work we investigate extending BPMN to model medical guidelines, as a vehicle for studying in a patient-focussed manner the interactions between treatments or interventions when the patient is suffering from multiple morbidities.

BPMN models are analysable, and suitable restrictions of the notation can be endowed with a strong mathematical semantics (e.g. [27, 28, 29, 30, 31, 32]). Most research has focused on the control flow aspects of BPMN, whereas analysis of data flow aspects of BPMN is a less studied topic — in BPMN the semantics of handling data remain unspecified and open to interpretation [33, 34]. In recent years modelling of data objects and mapping them to Petri nets has received considerable attention (e.g. [33, 35]): such research paves the way for analysis of BPMN models. However, to the best of our knowledge existing research focuses on single BPMN models. We address the question of detecting execution paths in two BPMN models which violate (or not) a set of logical constraints. For example, consider two BPMN models that represent two medical guidelines applied concurrently to a given patient. Medications are often prescribed through both pathways, but some of these medications cannot be prescribed simultaneously. How can we automatically detect the execution paths (treatment steps) that can use a combination of drugs with minimal conflict?

We present a method for detecting execution paths in two BPMN models that violate a given set of constraints. The outline of our approach is as follows: Firstly, we propose a data-enriched subset of BPMN, named BPMN+V, appropriate for modelling clinical guidelines and based on the Workflow Graphs proposed by Vanhatalo et al. [36]. Extending the semantics of Workflow Graphs to include data models, we obtain a data-rich execution semantics for our subset of BPMN. Secondly, we outline a transformation from BPMN+V to Coloured Petri Nets (CPN) [37] which provide a formalisation of BPMN+V to facilitate analysis. Thirdly, CPN models corresponding to two guidelines are composed and enhanced with logical constraints representing potential conflicts, such as drug incompatibility, between the associated data models. Finally, the state space analysis properties of CPN (such as deadlock detection) enable detection of conflicts in the composed model.

The paper is organised as follows. After reviewing background material in Section 2, in Section 3 we propose a simple data model for BPMN to facilitate the problem outlined in Section 4, of detecting conflict between care pathways. We describe the proposed approach in several parts in Section 5: the data-enriched BPMN model and associated semantics in Section 5.1 and following; mapping BPMN+V to CPN in Section 5.9; composition of clinical guidelines mapped as CPN, and detection of conflicts, in Sections 5.10 and 5.11. A case study (Section 2.1) is used to guide the discussion throughout the paper. The study uses two self-contained fragments of medical guidelines, for Osteoarthritis and COPD, which demonstrate conflicts between drug and lifestyle advice interactions. The final evaluation of the case study is in Section 6. Finally, in Section 7 we outline future research to extend this framework to enable many different types of conflict to be detected efficiently using logical analysis and constraint solvers such as Alloy [38] or Z3-SMT [39].

2. BACKGROUND

In this section, we present reference material on which our approach builds. We first describe our case study modelling conflicts between two care pathway fragments. We then give a brief background on business processes in general and clinical guidelines and care pathways in particular.

DOI: 10.1002/smr

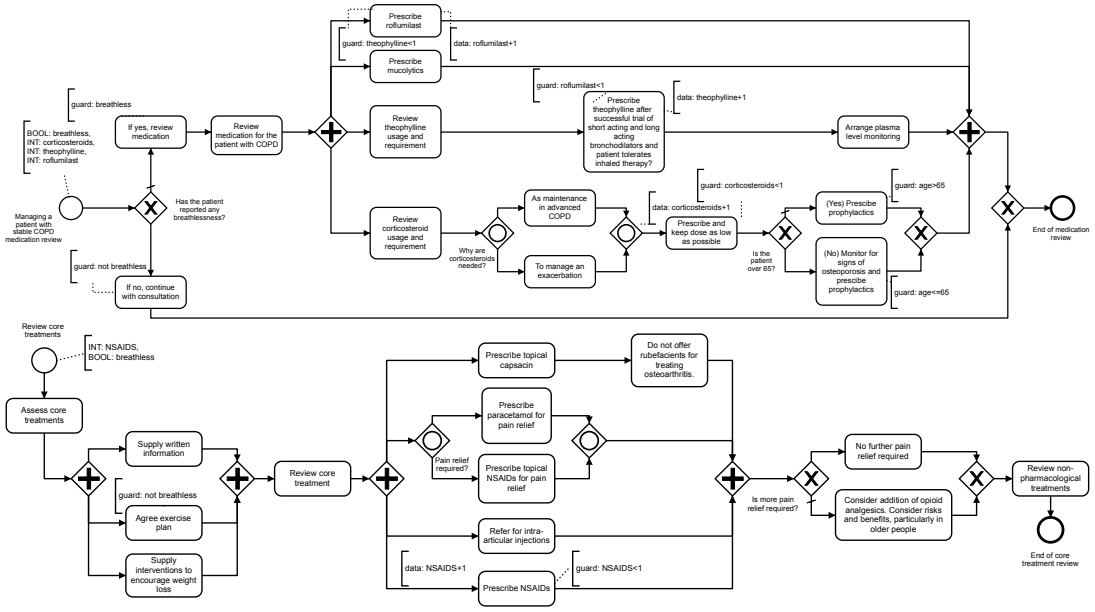


Figure 1. Pathway fragments for COPD (top) and Osteoarthritis (bottom) used in the case study, modelled as BPMN with annotations describing data interactions (BPMN+V).

Finally we review the Workflow Graph notation proposed by Vanhatalo et al. [36], which can be viewed as a subset of BPMN notation; and Coloured Petri Nets.

2.1. Case Study

To illustrate the proposed concepts and techniques, we have modelled two clinical care pathway excerpts using BPMN 2.0 notation [34]. These are illustrated in Fig. 1 and briefly described here. The review of medication depicted in the models would form part of a 10 minute review appointment with a General Practitioner (GP) in the UK, for patients with COPD [40] (top) and Osteoarthritis [41] (bottom). The pathways have been mapped from the NICE guidance as representative fragments of much larger models describing the treatment of these diseases.

COPD Medication Review: The review takes place only if the patient reports breathlessness. Four medications (roflumilast, mucolytics, theophylline and corticosteroids) are then reviewed in parallel, meaning there is no restriction on the order in which the reviews are carried out. In the case of theophylline, there are further criteria constraining prescription, and plasma level monitoring needs to be arranged. For corticosteroids, several further process steps are triggered. First, the reason for the drug requirement is established, then prescription is arranged within the constraint of keeping dosage as low as possible. Finally, dependent on the patient's age, prophylactics may be prescribed with or without monitoring for osteoporosis.

Osteoarthritis Treatment Review: This review proceeds in four stages: assess and then review core treatments, review pain relief, then review non-pharmalogical treatments. In the first stage, written information, the exercise plan, and weight loss interventions are reviewed together (no restriction on the order of the activities). The second stage involves the review of four treatments (topical capsacin, pain relief, intra-articular injections, and NSAIDs). The third stage assesses whether further pain relief is required, and indicates considerations if so.

If these pathways are followed concurrently, as when treating a patient diagnosed with both Osteoarthritis and COPD, then a conflict occurs. The British National Formulary [42] identifies that corticosteroids, which may be prescribed in the COPD pathway, and non-steroidal anti-inflammatory drugs (NSAIDs), which may be prescribed for Osteoarthritis, are in conflict. If

DOI: 10.1002/smr

prescribed together they may lead to serious complicating symptoms[§]. It is therefore of great importance that such interactions are discovered and avoided. Given that these example pathways are fragments of much larger care pathways extracted from the clinical guidelines, an automated method is crucial for doing so, and also for providing resolutions. Conflicts also exist between theophylline and roflumilast prescription in the COPD model, and potentially between the dependencies on 'breathlessness' in the two models.

2.2. Business Processes

Business processes describe activities carried out to fulfil a business function, and the relations between them [43], known as 'control-flow'. Such functions are varied, including service, financial or customer management, software development, or in healthcare, treatment of patients. Defining processes facilitates understanding, and managing the complex interactions between activities and resources can help demonstrate adherence to regulations, or increase efficiency. Various representational mechanisms have been suggested for capturing process control flow. These range from formal languages such as Petri nets [44, 45, 46] or BPMN [34] which allow systematic analysis and comparison, to flowchart notations used to informally discuss business processes, such as the care guidelines for COPD [40] or Osteoarthritis [41].

Petri nets are widely used to model concurrent and distributed systems, and are rigorously defined, whereas BPMN although standard and very flexible, is not fully formalised [47]. So-called Workflow models are therefore now widely used for specifying business processes [46, 48, 49, 50]. These restrict the structure and behaviour of the models to a subset adequate for specifying business processes with well-defined behaviour and properties for analysis. Focusing on the analysis of the systems, Van der Aalst et al. [46, 48] present a Workflow modelling language in which models are constructed from blocks of Petri net models representing common workflow constructs.

2.3. Clinical Guidelines and Care Pathways

Formation of clinical guidelines is complex [51, 52, 53], requiring critical appraisal of evidence from many sources such as systematic reviews and clinical trials. Care pathways distilled from the guidelines, used in clinical decision support systems, present as large and complex networks of many activities, typically broken down into interacting sub-processes (e.g. [40, 41]). Features of care pathways include steps for assessment, treatment and review, as well as links to related pathways. Diverse options may be given for treatment and advice to help improve a condition, including for example lifestyle recommendations, exercise specifications or dietary advice. Care pathways also detail medication options, prescribing the clinically appropriate medication both for the presenting complaint and also for prevention of other conditions for which risk factors have been noted.

Another major element within the pathway relates to specialist services referrals, which could be within secondary- or tertiary-centred care. After being referred, the patient will be under a different section of the pathway, specific to that healthcare sector. For example, some medications can only be prescribed under specialist rather than generalist care. All of these decisions will be noted within the pathway. Furthermore, the decisions have the ability to trigger different sections of the pathway, creating the need to go back and re-investigate earlier options within the path.

The implementation challenges introduced by such complex processes, and increasing time pressures within the NHS have promoted the need to look outside the healthcare sector for solutions. Computer Scientists have investigated novel solutions to problems associated with healthcare pathways and clinical guidelines [54], for example using string metrics to identify how patient journeys differ from the prescribed care pathway [55]. This is particularly useful for interrogation of the patient journey to identify real world management. Natural Language Processing (NLP) has also been used to help with clinical decision support systems [56], extract information from clinical guidelines [57] and electronic health records [58], and to test improvement to health such as a reduction of opioid prescribing [59].

[§] "Increased risk of gastro-intestinal bleeding and ulceration" [42].

DOI: 10.1002/smr

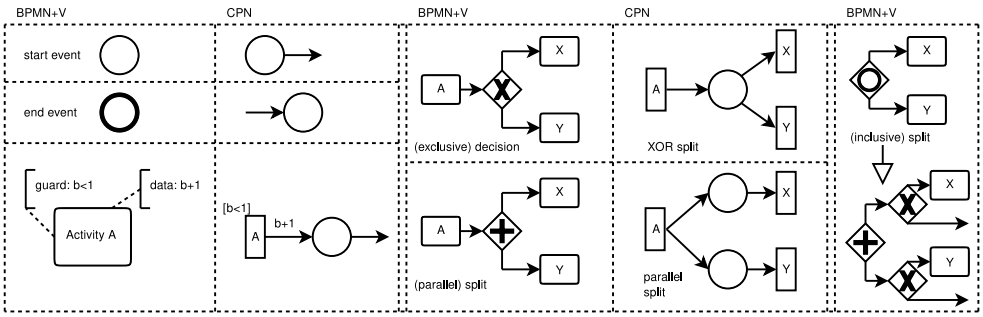


Figure 2. Main BPMN control-flow structures (left of each column pair), with annotations for BPMN+V (Section 5.1) and mappings to Coloured Petri Nets (Section 5.9, right of each column pair). Inclusive OR (far right column) splits are mapped as a combination of parallel and exclusive. Joins are mapped equivalently.

2.4. *BPMN for Modelling Clinical Pathways*

Many healthcare studies have employed BPMN to model clinical pathways (CPs) [4, 5, 6, 15, 16, 17]. Several authors have investigated the use of BPMN in healthcare [16, 17], and concluded that it is "sufficiently suitable for the planned modelling and imaging of CPs" [3], and its prevalence is increasing [16]. Key benefits of BPMN are stated as being graphically clear and appealing [3] and designed to facilitate communication between non-specialists [4, 5, 6]. BPMN has been criticised because its semantics are under-specified and models are not guaranteed to be interoperable between systems [60]. However it provides for extensions [16, 61], for example to use colour to enhance comprehensibility of complex clinical processes [5]. Formal semantics for subsets of BPMN [34] have been proposed including via transformation to Petri nets [47] or YAWL[¶] [29, 30, 31].

Formalisms such as Petri nets [44, 45, 46] and YAWL [62] do enjoy fully specified semantics, including (for YAWL) interaction with data. Arguably this is at the expense of the graphical clarity and interpretability of BPMN. They have also been criticised [60] as too restrictive for modelling 'real' business processes. Computer-Interpretable Guidelines (CIGs) have also been developed as part of clinical decision support systems (DSS) [63], such as PRO*forma* [64], Arden Syntax [65] and GLIF [66]. DSS provide a complete environment for process automation and decision support, including comprehensive supporting clinical information. The interaction with data of YAWL and PRO*forma* et al. is more complex than required for the conflict detection discussed here, and closely entwined with the process engine and decision support behaviour. Domain-specific languages may also risk limited acceptance and interoperability issues [67].

Since we are concerned here with modelling clinical guidelines in an accessible and human-interpretable way rather than process automation, we base our notation on BPMN for the benefits of widespread acceptance, ease of interpretation and extension described above. We hope that in the future our methods may be extended to other data-aware or domain-specific modelling languages.

2.5. *Workflow Graphs*

We find the modelling language suggested by Vanhatalo et al. [36], based on Petri nets, the closest to the style of BPMN 2.0 or major software development tools such as Oracle JDeveloper[‖]. Our model (Section 5.1) builds on their formalism, outlined next.

Definition 1
A *Workflow Graph* is a graph $G = (N, E)$, with a set of nodes N connected by a set of edges $E \subseteq N \times N$. Each node represents an activity or a control-flow construct from the set {START, STOP, ACTIVITY, FORK, JOIN, DECISION, MERGE}.

[¶]Yet Another Workflow Language [62].
[‖]http://www.oracle.com/.

DOI: 10.1002/smr

These relate to the core control-flow elements used in BPMN 2.0 (Fig. 2), with the exclusion of 'inclusive' gateways (a large circle O in a diamond), which we define in our formalism (Section 5.1). Small circles indicate START and STOP nodes; rectangles with rounded corners, ACTIVITY nodes; diamonds containing a large X, DECISION or MERGE; containing a large +, FORK or JOIN.

Notation 1
A Workflow Graph G is *well-formed* [47] by definition:

- G has a unique START node i_G with a single output edge and no input edges, and a unique STOP node o_G with a single input edge and no output edges;
- each node $n \in N$ has a set of input and output edges $I(n) \in E$ and $O(n) \in E$;
- each ACTIVITY node has a single input and a single output edge;
- each FORK and DECISION node has a single input edge and two or more outgoing edges; and
- each JOIN and MERGE node has two or more input edges and a single output edge.

An ACTIVITY node models an atomic (indivisible) unit of work. A DECISION node models choice between alternate sequences of activity following the node (MERGE nodes model the end of the alternate paths). A FORK node models sequences of activity which may happen in parallel following the node (JOIN nodes model the end of the parallel paths).

Definition 2
The *behavioural semantics* of Workflow Graphs is described as a 'token game', similar to the semantics of Petri nets [44, 45]. The flow of tokens through the graph indicates the progress of *instances* of the process execution. See Vanhatalo et al. [36] for full definitions.

Notation 2
A *state* s of a Workflow Graph $G = (N, E)$ is a mapping $s : E \to \mathbb{N}$ assigning tokens to edges E. We write $s(e) = k$ to indicate that in state s edge e carries $k \in \mathbb{N}$ tokens. An execution of node $n \in N$ results in changing the state of G from s to s', denoted $s \xrightarrow{n} s'$.

Informally, the state s of Workflow Graph G controls which nodes can be executed, while the semantics describe the change in state resulting from execution of a node. Execution of an node in G results in the movement of tokens between the edges to capture the flow of actions. An *instance* of the process (e.g. treatment review for a given patient) is started by executing the START node i_G, when a single token is added to its output edge $O(i_G)$. There is no change to the allocation of tokens to other edges in the graph. Multiple process instances may be executing concurrently, thus $O(i_G)$ may carry more than one token. Executing the STOP node o_G removes a token from its input edge, denoting completion of the instance. A node n of type {ACTIVITY, FORK, MERGE} can be executed if each input edge in $I(n)$ has one or more tokens. One token is then removed from each edge in $I(n)$, and one token is added to each edge in $O(n)$. If n is a DECISION node then a token is removed from the single $I(n)$, and added to just one output edge in $O(n)$, chosen non-deterministically. If n is a JOIN node, a single token is removed from one edge in $I(n)$, chosen non-deterministically from all those bearing one or more tokens, and passed to the single output edge $O(n)$.

2.6. Coloured Petri Nets

Coloured Petri Nets (CPN) [37] are commonly used to model concurrent systems and analyse their properties, particularly when data is involved. CPN extends the Petri net formalism with high-level programming language capabilities, enabling definition of data attached to the process and interaction between data and process behaviour. As such, they are appropriate for our need to analyse the interaction between multiple processes (care pathways), where process behaviour is strongly driven by data (patient characteristics).

Definition 3
A *Coloured Petri Net* is a bi-partite graph specified by a tuple $C = (P, T, A, \Sigma, V, S, G, E, l)$, where

- P is a finite set of *places*, T a finite set of *transitions* such that $P \cap T = \emptyset$, and $A \subseteq (P \times T) \cup (T \times P)$ a set of directed arcs. $N = (P, T, A)$ is the Petri net structure of C.

DOI: 10.1002/smr

- Σ is a finite set of *colour sets* describing types of data associated to the net; V is a finite set of *variables* taking on these types.
- $S : P \to \Sigma$ allocates colour sets to places.
- $G : T \to \text{EXPR}_V$ describes *guards* on transitions, defined using logical expressions or programming constructs, and interacting with data to determine when transitions can execute.
- $E : A \to \text{EXPR}_V$ describes guards on arcs, providing a data-driven mechanism to control the flow of tokens through the net.
- $I : P \to \text{EXPR}_\emptyset$ describes the initial state of the net, in terms of tokens in places.

Notation 3
The *behavioural semantics* of CPNs are governed by the flow through the net of data-enriched *coloured tokens*, which are associated with variables in V, controlled by the interaction of V with the types S of places and guard expressions G on transitions and E on arcs.

We use a subset of CPN to formalise our data-enriched BPMN (BPMN+V). We introduce some CPN concepts as necessary in later sections, and refer the interested reader to Jensen and Kristensen [37] for full definitions of the syntax and semantics of CPN. Suitably restricted subsets of BPMN and Petri nets can be defined with equivalent semantics, and a model transformation between them has been defined by Dijkman et al. [47]. We base a similar transformation on this, between BPMN+V and CPN, outlined in Section 5.9. CPN have been used as the formal target model for the underlying semantics of other models of concurrency, such as Sequence Diagrams (e.g. [68]).

3. A SIMPLE DATA MODEL FOR BPMN

Data is a core asset of any organisation, and plays a key role in business processes, forming the basis of many decisions. In the care pathway in Fig. 1 the decision to prescribe prophylactics or not in a COPD patient is taken on the basis of the age of the patient. In BPMN the primary construct for modelling data within the process flow is the *data object* element [34, p205]. However, the semantics of data objects remain unspecified and even left to the interpretation of the modellers [33, 34]. Moreover, in BPMN the behaviour of data objects is decoupled from the control-flow.

Data aspects of BPMN have received considerable attention recently [69, 70, 71]. Meyer et al. evaluate business process modelling languages with respect to data modelling and investigate modification of data by activities, events, gateways and control-flows [69]. They also present a set of algorithms for extracting such data models [70]. Sun et al. [71] present an elaborate formalism of the interaction between business processes and databases. They deal with crucial issues such as *isolation*, or when the execution of one process or instance must not interfere with that of another.

3.1. Data in Care Pathways

Each clinical guideline contains many tests, recommendations and actions covering all aspects of patient management for a specific condition. All involve data, whether pertaining to the patient characteristics, clinical aspects, or the environmental context. These data may guide the 'flow' of the patient through the care pathway (e.g. decisions taken on the basis of age or a blood test), and may themselves be modified by actions taken (e.g. necessity for further pain relief). For example, lifestyle advice may include data on minimum duration, frequency, and intensity of exercise, and depend upon patient age and specific symptoms; dietary advice includes examples; and medication recommendations depend on the presenting complaint but include consideration of cost and patient preference, as well as the context (generalist or specialist care) within which care is being provided. Data-driven decisions and data effects of activities may also trigger different sections of the pathway, or modify previous data attributes, causing earlier parts of the pathway to be revisited.

3.2. Modelling Care Pathway Data in BPMN

The Workflow Graph subset of BPMN we use for modelling the control-flow structure underling care pathways, corresponds closely to Workflow nets [36, 47]. It is therefore natural to draw on

DOI: 10.1002/smr

Coloured Petri Net (CPN) concepts to enrich our models with data. The formalism of coloured tokens carrying data attributes through the model, interacting with data-defined guards on elements of the net, is appropriate for modelling patients with given characteristics interacting with treatment activities in a particular clinical context. We assign to each token within a BPMN+V model a colour in the form of a vector of values, to each node a condition guarding execution of the node, and define *implicit* data which may affect the process execution.

Notation 4
Assume a fixed set of d variables $X = \{x_1, \ldots, x_d\}$ of types $\mathcal{T}(x_i) \in \{\mathcal{T}_1, \ldots, \mathcal{T}_m\}$. We model the data associated with a process instance as a d-tuple of valuations $V = (\nu_1, \ldots, \nu_d)$, assigning values to X as the business process executes and tokens flow through the net.

Data types include {boolean, integer, rational, enumerated (categorical)}. For example, in the COPD treatment process (Fig. 1) we might use boolean variables to record breathlessness and prescription of the various medications; enum. to record reasons for corticosteroid prescription and the age category of the patient; rational for plasma level; and integer for approximate weight.

Remark 1
Since we deal with well-defined models (Notation 1), the set of variables X is fixed and will not change. Our model is therefore a simplification of CPN.

Notation 5
A *condition* is a first order logic formula $c(\cdot)$ over variables in X. We define pre and $post$ conditions on nodes, as guards controlling when a node may be executed or may complete. For convenience we also define edge conditions equivalent to a pre condition on the following node:

Notation 6
A *node condition* on node n is a *pre-condition* $pre(n)$ such that $pre(n) = c(\cdot)$, controlling when n may be executed, or a *post-condition* $post(n)$ controlling when n may be considered complete. We write $c(\cdot) \vDash V$ if the variable valuation V satisfies $c(\cdot)$. For example, if n is an ACTIVITY a representing a treatment only valid for patients over 55 years of age, and we have a variable x_i indicating patient age in years, then $pre(a)$ would be $c(x_1, \ldots, x_d) \triangleq (x_i > 55)$.

Notation 7
An *edge condition* is a convenience when considering nodes with multiple output edges (i.e. diverging gateways). $c(e_{out})$ corresponds to the pre-condition of the node following the edge, i.e. $c(e_{out}) = pre(n) \mid n \in N \wedge I(n) = e_{out}$.

Notation 8
A *data modification* is a statement $f(\cdot)$ over variables in X describing the effect on data of the execution of an activity, e.g. $x := x + 1$ or $x := False$. We loosely define D as the set of all valid such data modification functions.

Notation 9
Assume that a process interacts with a database $\mathcal{D}$. We consider data in $\mathcal{D}$ to be *implicit* to the process, involved in defining the control-flow, but unchanged by it. For example, a database of drug interactions [42] describes constraints on the context in which a drug may be prescribed.

In Section 5.1 we describe an extension of the semantic model of Vanhatalo et al. by describing the changes in the value of colours as the tokens flow through the (data-enhanced) BPMN+V model. We believe our method will limit the anomalies between data and semantics (see e.g. [33]) through introducing a clear semantics which marries data and control flow. Explicit modelling of the interaction between data and control-flow enables a transparent view on how process execution is constrained by, and affects, patient characteristics, and implicitly depends on external attributes such as medication interactions and resource scheduling.

DOI: 10.1002/smr

4. PROBLEM DESCRIPTION

The problem we address is that of identifying conflicts between clinical care guidelines when they are followed concurrently in treating patients with multiple morbidities. Assume that we have two BPMN+V models M_1, M_2 representing two guidelines, and these interact with a database $\mathcal{D}$. Suppose further that each model has a set of relevant variables X_1, X_2 some of which are shared between the models. We collect the variables into a set $X = \{x_1, \ldots, x_d\} = X_1 \cup X_2$. For example, assume these variables represent the medication taken when a patient is treated by two pathways (*cf* the example in Section 2.1). If the patient is on medication m_i, then the corresponding variable x_i is set True. Knowledge of drug interactions (defined in $\mathcal{D}$) indicates that some combinations of medication are not permitted. We write this as a constraint over the variables.

Notation 10
Suppose that the medication related to variable x_i should never be taken with the the medication related to x_j, we write this as a *constraint* $C_r(x_1, \ldots, x_d) \triangleq \neg(x_i \wedge x_j)$. We assume that we know all k possible constraints on interactions between variables in X, $C = \{C_1, \ldots, C_k\}$. Although conflicts of this type between M_1 and M_2 could be identified by checking the values V assigned to X against the set of constraints C, it is more useful to pin-point which execution paths in the models are in conflict. We therefore pose the question in this form:

Question
Given two BPMN+V models M_1, M_2, variables X and a set of constraints C, identify all pairs of execution paths which will modify the variables in X so that at least one of the C is violated.

5. APPROACH

Our approach to detecting execution paths in conflict between two or more clinical guidelines follows several steps, detailed in the following sections:

1. define a process modelling language BPMN+V to capture control flow for clinical guidelines, and its dependencies and effect on data;
2. translate models described as BPMN+V to CPN to facilitate formal analysis;
3. compose CPN models representing multiple clinical guidelines, adding constraints between the models; and
4. solve the composed model using CPN state space methods to identify points of conflict such as deadlocks.

Remark 2
The translation to CPN is not strictly necessary, since the BPMN+V notation is consistent and complete for our purposes, but it allows us to take advantage of the existing theory of CPN analysis.

Remark 3
This is the first introduction of our method. In the future we plan to replace the CPN analysis with translation to logical constraints and efficient analysis using SAT and SMT solvers.

5.1. Semantics For BPMN With Data

We extend the Workflow Graph formalism ([36] and Section 2.5) to allow for the flow of data through the graph, using a form of colour for tokens. Our notion of data is defined in Section 3.2. We refer to this formalism as BPMN+V.

Definition 4
A *Workflow Graph extended with data (BPMN+V)* is a tuple $G = (N, l, E, X, pre, post, mod)$:

- N is a finite set of nodes (BPMN elements);
- $l : N \to \{\textsc{Start}, \textsc{End}, \textsc{Activity}, \textsc{Exclusive}, \textsc{Inclusive}, \textsc{Parallel}\}$ is a relation assigning each node a fixed type;

DOI: 10.1002/smr

- $E \subseteq N \times N$ is a finite set of edges (sequence flows) connecting nodes;
- $X = \{X_1, \ldots, X_d\}$ is a finite set of d variables associated with the process;
- $pre : N \to C$ is a set of *guards* defining node pre-conditions, as defined in Notation 5;
- $post : N \to C$ is a set of *guards* defining node post-conditions; and
- $mod : N \to D$ is a set of data modifications enacted by ACTIVITIES.

The execution semantics of BPMN+V are described by the flow of tokens through the model.

Notation 11
Let $m : E \to \{T_1, T_2, \ldots\}$ define a *marking* defining the *state* of the process, mapping each edge to a set of *coloured token* IDs. $|m(e)|$ defines the number of tokens on edge $e \in E$ in state m. Each token T_i is a pair (t_i, V_i), with $\tau(T_i) = t_i$ an ID unique to process instance i, $\mathcal{V}(T_i) = V_i$ a valuation of the variables X as defined in Notation 4. Executing node $n \in N$ changes the state from m to m', denoted $m \xrightarrow{n} m'$, as n consumes and produces tokens according to the semantics, and may modify the values V_i. Execution of a sequence of nodes is indicated by $m \xrightarrow{*} m'$.

Each instance of a process carries a single coloured token, which may be split by diverging PARALLEL nodes to describe parallel sequences of activity, and merged by converging PARALLEL nodes. The colour is synonymous with the assigned data values. The valuations affect process control flow by interacting with *guards* (pre and $post$) on nodes, and may be changed by executing ACTIVITY node mod statements, as the business process executes and tokens flow through the net.

Notation 12
In the following, unless otherwise indicated, T denotes a token carrying a d-tuple of valuations (colour) V before execution of the node under consideration. T' denotes the same token following execution, with possibly modified values V'.

5.2. START *Event*

A START event is a node $n \in N$ that captures the initialisation of a process instance, by creating a coloured token T' on its unique output edge. For example, a patient may be registered on a programme of treatment. Any BPMN+V model has exactly one START event. Assume V captures initial values of all data attributes associated with the process instance to be created, then n can execute (thus the process instance can start) if $pre(n) \vDash V$. The effect of executing $m \xrightarrow{n} m'$ is:

1. $V' = V$, and
2. $m'(e) = \begin{cases} m(e) \cup \{T'\} & \text{if } e \in O(n), \\ m(e) & \text{otherwise.} \end{cases}$
   Note that $m'(e) = m(e) \cup \{T'\} \Rightarrow |m'(e)| = |m(e)| + 1$.

5.3. END *Event*

An END event n brings the process to a conclusion, e.g. the patient no longer exhibits symptoms of the morbidity under consideration. Any BPMN+V model has exactly one END event. n can execute if $\exists T = (t, V) \in m\big(I(n)\big) \mid pre(n) \vDash V$. The effect of executing $m \xrightarrow{n} m'$ is:

1. $V' = V$, and
2. $m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e \in I(n), \\ m(e) & \text{otherwise} \end{cases}$
   Note that $m'(e) = m(e) \setminus \{T\} \Rightarrow |m'(e)| = |m(e)| - 1$.

5.4. *Sequence Flow*

Sequence Flow is represented by an edge e between two nodes, expressing flow of execution and carrying coloured tokens. Edge $e = (a, b)$ between nodes $a, b \in N$ is an *output edge* of its preceding node, $e \in O(a)$, and an *input edge* of its following node, $e \in I(b)$.

DOI: 10.1002/smr

5.5. ACTIVITY *Node*

An ACTIVITY node a represents an atomic (indivisible) unit of work to be executed. In a well-formed BPMN+V model an ACTIVITY has exactly one input sequence flow $e_{in} \in E$, i.e. $I(a) = \{e_{in}\}$, and exactly one output sequence flow $e_{out} \in E$, i.e. $O(a) = \{e_{out}\}$. On execution, a consumes a token $T = (t, V)$ from e_{in}, and returns $T' = (t, V')$ on e_{out}, with possibly modified assignment V'.

ACTIVITY a is executed if: $\exists T = (t, V) \in m(e_{in}) \mid pre(a) \vDash V$.

Execution of a changes the state such that $m \xrightarrow{a} m'$, where

1. $post(a) \vDash V'$, and
2. $$m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e = e_{in}, \\ m(e) \cup \{T'\} & \text{if } e = e_{out}, \\ m(e) & \text{otherwise.} \end{cases}$$

5.6. EXCLUSIVE *Gateway*

An EXCLUSIVE Gateway b models a decision point in a process. It can be either *diverging*: modelling a decision to be followed by one of several sequences of subsequent activity, or *converging*: modelling the rejoining of the alternative sequences created by the corresponding previous diverging EXCLUSIVE gateway.

Diverging: In a well-formed BPMN+V model a diverging EXCLUSIVE gateway b has one input sequence flow $e_{in} \in E$, such that $I(b) = \{e_{in}\}$, and two or more output sequence flows $O(b) \subset \{E \setminus e_{in}\}$. b consumes a token T from e_{in} and returns it unmodified to one of its output edges.

The gateway b executes if: $\exists T = (t, V) \in m(e_{in}) \mid pre(b) \vDash V$.

Then b is executed, $m \xrightarrow{b} m'$, and there exists a unique $e_{out} \in O(b)$ such that:

1. $V' = V$,
2. $c(e_{out}) \vDash V'$, and
3. $$m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e = e_{in}, \\ m(e) \cup \{T'\} & \text{if } e = e_{out}, \\ m(e) & \text{otherwise.} \end{cases}$$

A single output sequence flow can be followed after a diverging EXCLUSIVE Gateway b, subject to satisfaction of its edge condition (the pre-condition on the next node which will be executed, *cf* Notation 7). If the condition of more than one sequence flow after an Exclusive Gateway is satisfied, $|\{e_{out} | e_{out} \in O(b) \wedge c(e_{out}) \vDash V'\}| > 1$, then one output flow is chosen non-deterministically.

Converging: In a well-formed BPMN+V model, a converging EXCLUSIVE gateway b corresponds exactly to a preceding diverging EXCLUSIVE gateway b' with p output edges. Therefore b has p input sequence flows $|I(b) \subset \{E \setminus e_{out}\}| = p$. For a given process instance identified by token T with ID $\tau(T) = t$, a single edge $e_{in} \in I(b)$ can be active (carrying a token with ID t), i.e.

$$\forall e_{in} \in I(b), T = (t, V) \in m(e_{in}) \Rightarrow \forall e_i \in I(b) \wedge e_i \neq e_{in} \; (T_i \in m(e_i) \Rightarrow \tau(T_i) \neq t).$$

b has one output sequence flow $e_{out} \in E$, such that $O(b) = \{e_{out}\}$. b consumes a coloured token from e_{in} and returns it unmodified to e_{out}.

The gateway b therefore executes if

$$\exists e_{in} \in I(b) \wedge T = (t, V) \in m(e_{in}) \wedge \forall e_i \in I(b) \wedge e_i \neq e_{in} \wedge \forall T_i \in m(e_i) \; \tau(T_i) \neq t.$$

effecting $m \xrightarrow{b} m'$ such that

1. $V' = V$, and
2. $$m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e = e_{in}, \\ m(e) \cup \{T'\} & \text{if } e = e_{out}, \\ m(e) & \text{otherwise.} \end{cases}$$

DOI: 10.1002/smr

5.7. PARALLEL *Gateway*

A PARALLEL Gateway b indicates that all output sequence flows will be activated simultaneously.

Diverging: In a well-formed BPMN+V model, a diverging PARALLEL gateway b has one input sequence flow $e_{in} \in E$, such that $I(b) = \{e_{in}\}$, and two or more output sequence flows $O(b) \subset \{E \setminus e_{in}\}$. b consumes a token from e_{in} and duplicates it unmodified, to each output edge $e \in O(b)$.

The gateway b executes if: $\exists T = (t, V) \in m(e_{in}) \mid pre(b) \vDash V$ and $\forall e \in O(b), c(e) \vDash V'$.

Then $m \xrightarrow{b} m'$ such that:

1. $T' = T$, and
2. $m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e = e_{in}, \\ m(e) \cup \{T'\} & \forall e \in O(b), \\ m(e) & \text{otherwise.} \end{cases}$

Converging: In a well-formed BPMN+V model, a converging PARALLEL gateway corresponds exactly to a preceding diverging PARALLEL gateway b' with p output sequence flows. Therefore b has p input sequence flows, $|I(b) \subset \{E \setminus e_{out}\}| = p$, and one output sequence flow $e_{out} \in E$, such that $O(b) = \{e_{out}\}$.

Since each input edge of b concludes a particular concurrently executing sequence of activities following b', the data assignments V_i carried by tokens $T_i = (t, V_i)$, arriving at b on edges $e_i \in I(b)$ for a given process instance, may differ. For b to execute, these assignments must be *compatible*. They must then be synchronised to a single assignment V' on e_{out}.

Notation 13
Compatible data assignments $V_1, \ldots, V_d$ on tokens arriving at b are defined with respect to the assignment Y on tokens leaving b'. Let the data assignments to tokens leaving b', arriving at b, and leaving b, be denoted respectively:

- each $e'_{out} \in O(b')$ carries token $S = (t, Y)$ and $Y = (y_1, \ldots, y_d)$,
- $e_j \in I(b)$ carries token $T = (t, V_j)$ and $V_j = (\nu_1^j, \ldots, \nu_d^j)$, $0 < j \leq p$, and
- e_{out} carries token $T' = (t, V')$ and $V' = (\nu'_1, \ldots, \nu'_d)$.

Then $Y, V_1, \ldots, V_p, V'$ satisfy one of the following three criteria; $\forall 0 < i \leq d$,

1. $\forall 0 < j \leq p, \nu_i^j = y_i$; assignment to variable x_i is not changed by any parallel path following b': we set $\nu'_i = y_i$;
2. $\exists 0 < j \leq p, \nu_i^j \neq y_i \wedge \forall 0 < k \leq p, k \neq j, \nu_i^k = y_i$, assignment to x_i is changed on one parallel path only: we set $\nu'_i = \nu_i^j$; or
3. $\exists 0 < j \leq p, \nu_i^j \neq y_i \wedge \exists 0 < k \leq p, k \neq j, \nu_i^k \neq y_i$, assignment to x_i is changed on more than one parallel path, the data cannot be synchronised, and the gateway cannot execute.

In the case 1. and 2. the differing data assignments on each input are *compatible* with each other and with Y, denoted $compat(V_1, \ldots, V_p, Y)$, and can be synchronised by setting the elements of V' as stated. We can relax the condition of equality of assignments to a suitable definition of approximate equality such as numeric values within some threshold. In case 3. the data have been changed incompatibly and cannot be synchronised: the gateway cannot execute.

Therefore b can execute when each input sequence flow $e_{in} \in I(b)$ has a token with the same ID and the data assignments are compatible, i.e.

$$\forall e_{in} \in I(b)\, T_i = (t, V_i) \in m(e_{in}) \wedge \forall e_{out} \in O(b')\, S = (t, Y) \in m(e_{out}) \wedge compat(V_1, \ldots, V_p, Y).$$

b consumes a token from each $e \in I(b)$, $m \xrightarrow{b} m'$, and creates a single token on e_{out}, such that:

1. $V' = (\nu'_1, \ldots, \nu'_d)$ s.t. ν'_i are assigned according to compatibility cases 1. and 2. above, and
2. $m'(e) = \begin{cases} m(e) \setminus \{T_i\} & \text{if } e \in I(b) \wedge T_i = (t, V_i) \in m(e), \\ m(e) \cup \{T'\} & \text{if } e = e_{out}, \\ m(e) & \text{otherwise.} \end{cases}$

DOI: 10.1002/smr

5.8. INCLUSIVE *Gateway*

An INCLUSIVE Gateway b models a decision in a process and can trigger more than one outgoing sequence flow. An INCLUSIVE gateway can be either converging or diverging. It is a combination of EXCLUSIVE and PARALLEL gateways; a subset of outgoing sequence flows can be activated following a diverging gateway, paths that are activated take place in parallel, and all and only those paths that are activated must be synchronised at the subsequent converging gateway.

Diverging: In a well-formed BPMN+V model, a diverging INCLUSIVE gateway b has one input sequence flow $e_{in} \in E$ s.t. $I(b) = \{e_{in}\}$, and two or more output sequence flows $O(b) \subset \{E \setminus e_{in}\}$. b consumes a token from e_{in} and duplicates it unmodified to a subset of its output edges $E_O \subset O(b)$.

The gateway b executes if: $\exists T = (t, V) \in m(e_{in}) \mid pre(b) \vDash V$. Then $m \xrightarrow{b} m'$, such that:

1. $V' = V$, and
2. $m'(e) = \begin{cases} m(e) \setminus \{T\} & \text{if } e \in I(b), \\ m(e) \cup \{T'\} & \text{if } e \in O(b) \wedge c(e) \vDash V', \\ m(e) & \text{otherwise.} \end{cases}$

Converging: If b is converging, then $I(b) \subset E$ and $O(b) = \{e_{out}\}$. In a well-formed BPMN+V model, the subset $E_I \subset I(b)$ of input sequence flows to a converging INCLUSIVE gateway, receiving tokens for a given process instance ID t, must correspond with the subset $E'_O \in O(b')$ of outgoing sequence flows from the preceding diverging INCLUSIVE gateway on which tokens with ID t were created. We write $E_i \equiv E'_O$ iff $|E_I| = |E'_O|$ and

$$\wedge \forall e_i \in E_I \; \exists e_o \in E'_O, T_o \in m(e_o) \wedge \tau(T_o) = t \wedge m \xrightarrow{*} m' \wedge T_i \in m'(e_i) \wedge \tau(T_i) = t,$$
$$\wedge \forall e_o \in E'_O \; \exists e_i \in E_I, T_i \in m'(e_i) \wedge \tau(T_i) = t \wedge m \xrightarrow{*} m' \wedge T_o \in m(e_o) \wedge \tau(T_o) = t.$$

The gateway b executes if $E_I \equiv E'_O$, and data assignments carried by tokens on E_I are compatible (Notation 13). Then b consumes a token from each $e \in E_I$, $m \xrightarrow{b} m'$, and a single token is created on e_{out}, such that:

1. $V' = (\nu'_1, \ldots, \nu'_d)$ s.t. ν'_i are assigned according to compatibility cases 1. and 2. above, and
2. $m'(e) = \begin{cases} m(e) \setminus \{T_i\} & \text{if } e \in E_I \wedge T_i = (t, V_i) \in m(e), \\ m(e) \cup \{T'\} & \text{if } e = e_{out}, \\ m(e) & \text{otherwise.} \end{cases}$

5.9. *Transformation From BPMN+V to CPN*

BPMN+V supports only a subset of the semantics of CPN, similar to the restriction of BPMN and Petri nets supported by Workflow Graphs. The transformation from BPMN to Petri nets is complicated by the relatively informal semantics of BPMN and features such as INCLUSIVE gateways, events and message passing, none of which are directly supported by CPN. However, many INCLUSIVE gateway structures can be modelled by combinations of EXCLUSIVE and PARALLEL gateways (e.g. [72, 73]). Events and messages are currently not part of our BPMN+V specification. We therefore use a subset of the transformations described by Dijkman et al. [47] for Place-Transition (non-Coloured) Petri nets, to transform the BPMN control structures defined in the previous sections to CPN structures. We extend them to map BPMN+V conditions and data modification statements onto CPN transition guards and arc inscriptions, as summarised below.

- START and END events are mapped to a start and end place respectively.
- ACTIVITY nodes map to CPN transitions with pre- and post-places as necessitated by the surrounding nodes. Associated pre-conditions are mapped to transition guards, post-conditions and data modifications to inscriptions on the outgoing arcs.
- PARALLEL and EXCLUSIVE gateways map to parallel and exclusive-or (XOR) splits whose semantics are governed by the appropriate use of CPN places (Fig. 2).

DOI: 10.1002/smr

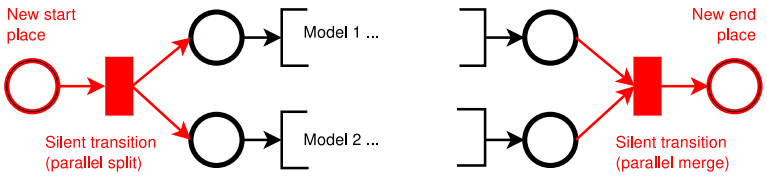


Figure 3. Parallel composition of two care pathway models.

- INCLUSIVE gateways are modelled using a combination of parallel and exclusive splits, allowing each sequence of activities following the split to be either executed or bypassed.

5.10. Composition of CPN Models

In this work, we employ a simple parallel composition. This is achieved by preceding the start places of the two models by a 'silent' CPN transition and a new start place, and following the end places with a second 'silent' transition and new end place (Fig. 3). This allows the two models to be followed simultaneously, simulating a patient following two care pathways. This makes the simplifying assumption that a patient starts both pathways at the same time, whereas most likely they would be following one or more pathways when a new one is started. However this approach ensures maximum overlap between the models and thus for all conflicts to be detected. The method could be extended to allow models to be connected at any point, to apply to a specific situation.

5.11. Conflict Detection

To support the evaluation of the BPMN+V model, we use state-space analysis [37] techniques to identify conflicts introduced by the composition of the CPN models. The main purpose is to demonstrate the suitability of BPMN+V to describe care pathways and allow conflicts to be detected.

To simplify the evaluation, we restrict our attention to three types of conflict, described next in relation to the artificial care pathway fragments illustrated in Table I.

Data assignment. A type of conflict, or inconsistency within a model, can be created by interaction between the data valuation V associated with a process instance, and the condition on a single ACTIVITY a. For example, Table I model (1) will be blocked if the patient is already prescribed with NSAIDs since ACTIVITY "Prescribe NSAIDs" cannot execute. A resolution might be to introduce a bypass, as in models (7) and (8).

Single variable. If a pair of ACTIVITIES a, b in eiher a single or pair of models check and modify the same variable x in similar ways, they may conflict. Models (3) and (4) in Table I for example check and set 'breathless' in opposing ways. Similarly, duplicating prescription of medication [74, 75, 76, 77] or tests [78, 79, 80] could both adversely affect patient safety and efficiency of the healthcare practice.

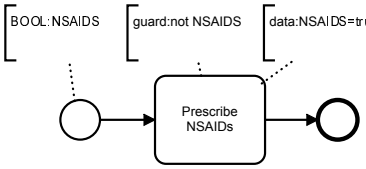
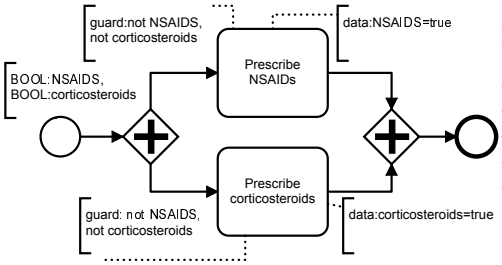
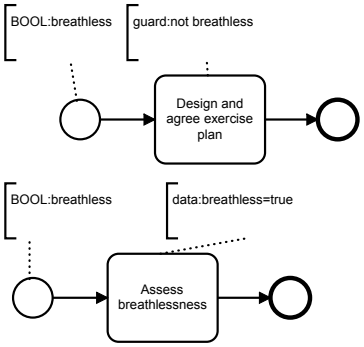
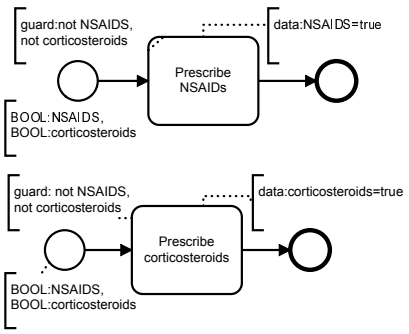
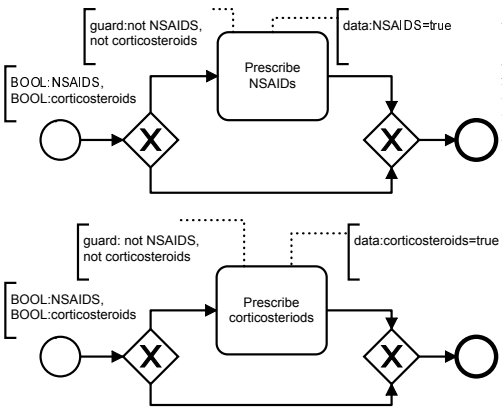
Pairs of variables. If ACTIVITY a has a pre-condition on variable x_1 and modifies x_2, while ACTIVITY b has a pre-condition on x_2 and modifies x_1, they may be in conflict. For example, two drugs which may not be prescribed together [42]: *cf* models (5) and (6) in Table I.

Conflict detection proceeds in two phases. We first analyse the individual care pathway models using the steps described below, to determine inconsistencies preventing either model from executing under certain data valuations. The models are then composed and the same analysis applied, excluding tokens with valuations which failed the single models.

1. Identify the d variables X involved in on pre- and post-conditions, and the values V to be checked or assigned, e.g. condition $x < 1$ involves variable x and value 1.
2. Create a 'covering set' of 2^d coloured tokens for which the model needs to be checked. For each $x_i \in X$, construct the pair of valuations $\nu_i \in V$ that will satisfy and fail each pre- and post-condition. E.g. tokens must include both True and False values for conditions x or $\neg x$; and with values to satisfy both $y > 1$ and $y \leq 1$, for an integer condition $y > 1$.

DOI: 10.1002/smr

Table I. Artificially-designed illustrative care pathway fragments to demonstrate conflicts within and between models (Section 6.1 and results in Table II).

Example Model(s)	Description
(1) 	Single model exhibiting a problem with certain data settings. Non-steroidal anti-inflammatory drugs (NSAIDs) are prescribed but should not be over-prescribed. The associated variable is set to True on prescription, but checked first to avoid over-prescription.
(2) 	Single model in which two activities are in conflict. Both NSAIDs and corticosteroids are prescribed (in parallel). These drugs should not be prescribed together nor individually over-prescribed, hence the guards. (Note that whereas in this illustration, these guards have been added manually, in reality they would be automatically discovered, e.g. from the BNF database [42].)
(3) & (4) 	Two care pathway fragments which exhibit conflict when combined. The first model includes prescription of an exercise plan, which should be avoided if the patient exhibits breathlessness. The second model fragment sets a variable indicating breathlessness, which will be in conflict.
(5) & (6) 	Two pathway fragments which combine the drug conflict of example model (2) with between-model conflict illustrated in example models (3) and (4). The guards and data modifications of the two prescription activities prevent the model from executing.
(7) & (8) 	An example of one way in which example models (5) and (6) might be changed to avoid conflict. The prescription activities can now be bypassed. However, the conflict detection technique presented will not discover the *potential* for conflict between these activities.

DOI: 10.1002/smr

3. Construct the reachability graph R_i $(1 \leq i < 2^d)$ of the CPN transformation of the model, corresponding to each token.
4. Identify dead markings in all R_i, and the corresponding blocked ACTIVITIES, conditions which failed, and the token data assignment and condition variables which are in conflict.
5. A pair of dead markings linked by common variables indicates that the corresponding ACTIVITIES conflict. The dead markings result from different paths through the model. For example, in Table I model (4), one path through leads to "Prescribe NSAIDs" before "Prescribe corticosteroids", resulting in the latter being blocked. An alternate path leads to the two activities in the reverse order, with "Prescribe NSAIDs" being blocked. This will result in two dead markings connected by the boolean variables 'NSAIDS' and 'corticosteroids'.
6. A single dead marking indicates a conflict due to initial data settings, such as an attempt to prescribe an already-prescribed medication.

We evaluate this approach in the next section. Section 7 outlines plans for more efficient analysis and determination of medically-appropriate conflicts.

6. EVALUATION

In this section we describe a three stage empirical evaluation of the modelling language and approach described in Section 5**. We first apply the approach to artificially-designed care pathway fragments exhibiting conflicts, to validate that BPMN+V is sufficient and complete for modelling care pathways, and that the approach for finding conflicts between them is effective. We next investigate the efficiency and scalability of the method for conflict detection, using large numbers of randomly-generated synthetic models of various size and complexity, with randomly added conflicts. Finally we apply our method to the case study (Section 2.1), to show that the known conflicts between the medical guidelines for COPD and Osteoarthritis are successfully identified.

To facilitate the evaluation we use BPMN annotations with structured text to specify the data aspects of BPMN+V, as illustrated in Fig. 2. Variables are specified as an annotation to the START event, ACTIVITY pre-conditions and data modifications are specified by annotations beginning 'guard:' and 'data:' respectively. We do not specify any post-conditions. These restrictions are not fundamental and in future work we will develop approaches more friendly to the modeller.

6.1. Artificial Care Pathway Fragments

In Table I we illustrate artificially-designed care pathway fragments to validate the BPMN+V modelling language and the state space method for conflict detection. These are not intended to be complete standalone pathways but rather to illustrate realistic conflict situations due to (e.g.) potential over-prescription [74, 75, 76, 77]. The annotations for data dependencies and modification have been added manually, whereas in the future such constraints will be automatically discovered using references such as the British National Formulary [42]. As described in Table I, these models exhibit various data-related inconsistencies and conflicts between models, explained in Section 5.11.

The results of the conflict detection method are shown in Table II. The 'Activity' column indicates the ACTIVITY a found to be blocked or in conflict. 'Data' reports the valuation which caused the conflict, 'Initial Data' the valuation of the initial token. 'Conflict Model', 'Conflict Activity' and 'Conflict Data' list any activities and variables found to be in conflict with a. These being empty indicates that the conflict is caused by the initial token valuation.

For model (1), we correctly identify that the model will be blocked for a patient who is already prescribed with NSAIDs. Similarly, Model (2) is blocked when either NSAIDs or corticosteroids are already prescribed. These activities are also in conflict. Since they occur in parallel, either may occur first and set the variable indicating prescription of the relevant medication, which conflicts with the guard on the other activity. A similar situation occurs between 'roflumilast' and 'theophylline' in the

**Code and results can be found at https://bitbucket.org/uobmitcon/jsep2017.

DOI: 10.1002/smr

Table II. Results of applying the state-space based conflict detection method to artificial models (Table I and Section 6.1).

Model	Activity	Data	Initial Data	Conflict Model	Conflict Activity	Conflict Data
1	Prescribe NSAIDs	NSAIDS=True	NSAIDS=True			
2	Prescribe NSAIDs	NSAIDS=True	NSAIDS=True			
2	Prescribe corticosteroids	CS=True	CS=True			
2	Prescribe NSAIDs	CS=True	CS=True	2	Prescribe NSAIDs, Prescribe corticosteroids	NSAIDS
2	Prescribe corticosteroids	NSAIDS=True	NSAIDS=True	2	Prescribe NSAIDs, Prescribe corticosteroids	CS
2	Prescribe NSAIDs	CS=True	CS=False	2	Prescribe corticosteroids	NSAIDS
2	Prescribe corticosteroids	NSAIDS=True	NSAIDS=False	2	Prescribe NSAIDs	CS
3	Design and agree exercise plan	breathless=True	breathless=True			
3	Design and agree exercise plan	breathless=True	breathless=False	4		
5	Prescribe NSAIDs	NSAIDS=True	NSAIDS=True			
5	Prescribe NSAIDs	CS=True	CS=True			
6	Prescribe corticosteroids	CS=True	CS=True			
6	Prescribe corticosteroids	NSAIDS=True	NSAIDS=True			
5	Prescribe NSAIDs	CS=True	CS=False	6	Prescribe corticosteroids	NSAIDS
6	Prescribe corticosteroids	NSAIDS=True	NSAIDS=False	5	Prescribe NSAIDs	CS
7 & 8	*No conflicts detected*					

case study COPD example (Fig. 1). Model (3) is correctly identified as blocked when 'breathless' is True (exercise should not be prescribed) but although the conflict with the setting of this variable in model (4) has been detected, it could not be fully identified. Further work is needed to determine if this is a problem with the implementation of the method, or a feature of this artificial example. Finally, models (5) and (6) exhibit the same medication conflict as model (2), illustrating that these conflicts are detected when two care pathways are combined. Models (7) and (8) illustrate that a recommendation to resolve these conflicts might be to introduce a bypass for these tasks. This could be implemented as a series of steps to check medication and prescribe only if appropriate, mitigating the danger of unsafe prescription. For these modified models, no conflicts are detected, raising the question of whether an extended method is desirable to highlight such potential conflicts.

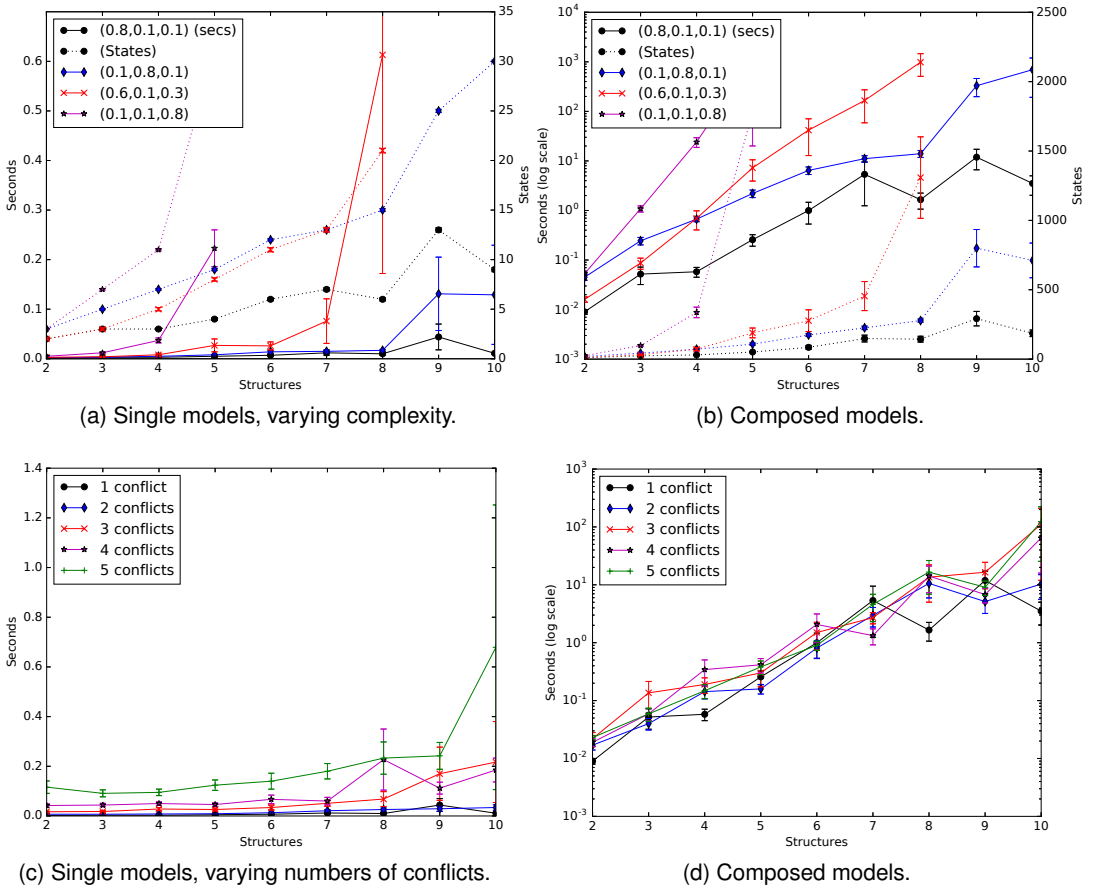
6.2. *Efficiency and Scalability*

To investigate the performance of the method for conflict detection we generated pairs of random models of increasing size and complexity, similarly to the method described for Process Log Generation [81]. Varying numbers of conflicts were inserted into one or both models. Models were generated by starting with a START–ACTIVITY–END sequence (e.g. model (1), Table I) and repeatedly randomly choosing an ACTIVITY to expand, up to a given number of structures in the target model. The chosen ACTIVITY was randomly replaced with either a sequence of two ACTIVITIES, an EXCLUSIVE diverging gateway followed by two ACTIVITIES and an EXCLUSIVE merge, or a PARALLEL gateway followed by two ACTIVITIES and a PARALLEL merge. We experimented with various probabilities for selection of each type of structure.

Fig. 4 reports statistics collected over 30 models of each type. Figs. 4(a) and 4(b) show that while the time for processing increases somewhat with the number of activities in any model, it is particularly sensitive to the amount of parallelism in the model. Fig. 4(b) shows that for composed models with a large amount of parallel activities in the original models, the time for conflict detection increases approximately exponentially with the number of structures. This is not unexpected, since the more activities are in parallel, the faster the size of the state space increases. (We do not consider models with cycles, so the state space remains finite.)

The COPD and OA examples (Fig. 1) each contain 14 activities. These are formalised from NICE pathways [40, 41] presented as hierarchies of 5-10 flowcharts of approximately 10 activities each, suggesting full models could have at least 50 activities. Studies have shown that 16% of adult

DOI: 10.1002/smr



(a) Single models, varying complexity.

(b) Composed models.

(c) Single models, varying numbers of conflicts.

(d) Composed models.

Figure 4. Performance of the state space conflict detection method (Section 6.2) for models of varying size and complexity, averaged over 30 randomly generated models. Top row (a,b): number of states and time (seconds) for conflict detection, for models generated with varying probability of creating sequence, XOR or parallel split and join structures. $(0.8, 0.1, 0.1)$ indicates $p(seq) = 0.8, p(xor) = 0.1, p(parallel) = 0.1$, etc. Bottom row (c,d): mean time for increasing numbers of randomly-generated conflicts, in models with mostly sequential structures (low parallelism). Left column (a,c): statistics for single models. Right column (b,d): statistics for the composed models.

patients have more than one chronic condition [82, 83], increasing to 65% of over 65s with two or more [84]. Although performance efficiency is thus a potentially serious problem for the state space method described here, especially since the composed models are inherently parallel, these results were obtained using a basic state space calculation algorithm. As discussed in Section 7, we plan to use more efficient logical analysis methods. The suitability of the BPMN+V modelling language is not affected by these results.

Although many conflicts may exist between models – Dumbreck et al. [77] found that between 89 and 133 drug-drug interactions were possible between the guidelines for three conditions in combination with up to 11 other co-morbidities – Figs. 4(c) and 4(d) show that the method is less affected by the numbers of conflicts introduced into the model.

6.3. Case Study

In Table III we report the conflicts automatically identified for the case study introduced in Section 2.1. The top and centre sections report for the OA and COPD models individually; the lower section for the composed model. In this study we used integer variables for medications instead of boolean, to indicate the number of times a medication was prescribed.

DOI: 10.1002/smr

Table III. Results of applying the state-space conflict detection method to the COPD and OA case study (Sections 2.1 and 6.3). CS abbreviates 'corticosteroids', RF 'roflumilast' and TH 'theophylline'.

Model	Activity	Data	Initial Data	Conflict Model	Conflict Activity	Conflict Data
OA	Agree exercise plan	breathless=True	breathless=True			
OA	Prescribe NSAIDs	NSAIDS=1.0	NSAIDS=1.0			
COPD	Prescribe and keep dose ...	CS=1.0	CS=1.0			
COPD	Prescribe RF	TH=2.0	TH=1.0			
COPD	Prescribe TH after successful ...	RF=2.0	RF=1.0			
COPD	Prescribe RF	TH=1.0	TH=0.0	COPD	Prescribe TH after ...	RF
COPD	Prescribe TH after successful ...	RF=1.0	RF=0.0	COPD	Prescribe RF	TH
OA	Prescribe NSAIDs	NSAIDS=1.0	NSAIDS=1.0	COPD		
COPD	Prescribe and keep dose ...	CS=1.0	CS=1.0	OA		
OA	Prescribe NSAIDs	CS=1.0	CS=1.0	COPD	Prescribe and keep ...	NSAIDS
COPD	Prescribe and keep dose ...	NSAIDS=1.0	NSAIDS=1.0	OA	Prescribe NSAIDs	CS

For OA and COPD the data valuations which would block the model were correctly identified, namely the parallel ACTIVITY "Agree exercise plan" in OA is blocked if the patient reports breathlessness, and NSAIDs cannot be prescribed if already being taken. Similarly for COPD, the activities to prescribe theophylline and roflumilast block execution of the model if the patient is already taking these medications. Theophylline and roflumilast may not be taken together, and this conflict within the COPD model was also discovered correctly.

To obtain a meaningful analysis of the composed model we removed the 'breathless' constraint from OA, since if True it blocks the OA part of the model, while if False, the COPD model is bypassed. We also added bypasses for the theophylline and roflumilast activities for similar reasons. A natural extension of this work would be automatically recommend model changes to resolve conflicts; these modifications could be regarded as implementing such recommendations.

As shown in Section 6.2 our state space exploration became impractical for large numbers of states. The reachability graph for the OA and COPD composed model contains over 11,000 states for some data assignments. Therefore we manually transferred the composed CPN model to CPN/Tools [85, 86] for state space analysis, and the dead markings back to our tool for analysis. Such an interface could be automated to allow efficient automatic state space exploration. The conflict between NSAIDs and corticosteroid prescription was successfully discovered.

7. CONCLUSION AND FUTURE WORK

We have presented BPMN+V, a data-enriched subset of the Business Process Model and Notation [1] (BPMN) suitable for modelling clinical guidelines. We defined a semantics for BPMN+V, based on Workflow Graphs [36] and Coloured Petri Nets [37], which allows the effect of data upon the guideline, and of the guideline upon the data, to be formally described. For instance, we can specify how the valuations assigned to data attributes, such as the medications prescribed to a patient, control the execution of activities, and how executing an activity (such as prescribing a medication), modifies that data. We then evaluated this model using a state space analysis approach to detecting the execution paths in two BPMN models which violate a given set of constraints.

The evaluation applied the method to artificial models, and to a real life case study using parts of the clinical guidelines for treatment of Chronic Obstructive Pulmonary Disease (COPD) and Osteoarthritis (OA). Using models designed in BPMN+V, the known conflicts were discovered successfully using the state space method. Performance analysis however showed the method to be impractical for larger models, especially those containing a high degree of parallelism (as would often be the case in a non-prescriptive guideline). This may be addressed by integrating state of the art state space analysis algorithms such as ASAP [87] or as implemented in CPN/Tools [86].

DOI: 10.1002/smr

In future work we plan rather to avoid state space methods, by translating BPMN+V models into logical constraints (see e.g. [88]) to allow efficient analysis using SAT and SMT solvers such as Alloy [38] or Z3-SMT [39]. This will allow much larger models to be efficiently analysed and formally proved correct and complete. Whereas in the evaluations in Section 6 we designed the constraints into the example models, in the future relevant constraints will be automatically discovered and added to the models, using references such as the British National Formulary [42] for medication conflicts. In consultation with clinical experts, we will define a comprehensive set of potential conflicts, and methods to detect and recommend changes to mitigate them. These conflicts will include medication and lifestyle, as discussed here, but also scheduling problems relating to availability of resources such as appointments, medical personnel, and locations. Finally we plan to build the methods into a tool suitable for use by clinicians, necessitating development of a suitable user interface and natural language methods for inferring data and constraints from text.

ACKNOWLEDGEMENT

This research was funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/M014401/1.

REFERENCES

1. T. Allweyer and D. Allweyer. *BPMN 2.0 Business Model and Notation. Einführung in den Standard für die Geschäftsprozessmodellierung*, 2010.
2. International Organization for Standardization, "ISO/IEC 19510:2013 Information technology – Object Management Group Business Process Model and Notation", 2013.
3. H. Scheuerlein, F. Rauchfuss, Y. Dittmar, R. Moller, T. Lehmann, N. Pienkos, and U. Settmacher, "New methods for clinical pathways – Business Process Modeling Notation (BPMN) and Tangible Business Process Podeling (t.BPM)," *Langenbeck's Archives of Surgery*, 397(5):755–761, 2012.
4. F. Ruiz, F. Garcia, L. Calahorra, C. Llorente, L. Gon, C. Daniel, and B. Blobel, "Business process modeling in healthcare," *Studies in Health Technology and Informatics*, 179:75–87, 2012.
5. R. Müller and A. Rogge-Solti. BPMN for healthcare processes. In *Proc. 3rd Central-European Workshop on Services and their Composition, Services und ihre Komposition, ZEUS 2011, Karlsruhe, Germany*, 705:65-72, CEUR-WS.org, 2011.
6. M. G. Rojo, E. Rolón, L. Calahorra, F. O. García, R. P. Sánchez, F. Ruiz, N. Ballester, M. Armenteros, T. Rodríguez, and R. M. Espartero, "Implementation of the Business Process Modelling Notation (BPMN) in the modelling of anatomic pathology processes," *Diagnostic Pathology*, 3(Suppl 1):S22, 2008.
7. M. Juric, B. Matthew, and P. Sarang. *Business Process Execution Language for Web Services: BPEL and BPEL4WS*. Packt Publishing, 2004.
8. P. Harmon. The state of business process management 2016. BPTrends, 2016.
9. J. Recker. Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal*, 16(1):181-201, 2010.
10. R. T. de Sousa, F. E. G. de Deus, B. A. de Sousa, A. P. F. Araújo, M. Holanda, W. M. C. Silva, H. Freitas, S. S. A. N. Vidal, R. M. G. dos Santos, and A. Moraes. A methodology for quality assurance for business process modeling with BPMN: A case study for the SIGEPE software. In *Proc. 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1-5, Piscataway, NJ, USA, 2016.
11. A. Herden, P. P. M. Farias, and A. B. Albuquerque. An Agile approach to improve process-oriented software development. In *Proc. Software Engineering Perspectives and Application in Intelligent Systems*, *Advances in Intelligent Systems and Computing*, 465:413-424. Springer, 2016.
12. M. zur Muehlen and D. T. Ho. Service process innovation: A case study of BPMN in practice. In *Proc. 41st Hawaii International International Conference on Systems Science (HICSS-41), Waikoloa, Big Island, HI, USA*, pp. 372-381. IEEE Computer Society, 2008.
13. E. Alreshidi, M. Mourshed, and Y. Rezgui. Cloud-based BIM governance platform requirements and specifications: software engineering approach using BPMN and UML. *Journal of Computing in Civil Engineering*, 30(4):04015063-1-23, 2015.
14. B. S. Barn and S. Oussena. BPMN, toolsets, and methodology: A case study of business process management in higher education. In *Proc. Information Systems Development, Towards a Service Provision Society (ISD)*, Paphos, Cyprus, pp. 685-693. Springer, 2008.
15. F. Zerbato, B. Oliboni, C. Combi, M. Campos, and J. M. Juarez, "BPMN-based representation and comparison of clinical pathways for catheter-related bloodstream infections," in *International Conference on Healthcare Informatics, ICHI, Dallas, TX, USA, 2015* pp. 346–355, IEEE Computer Society, 2015.
16. R. Braun, H. Schlieter, M. Burwitz, and W. Esswein, "BPMN4CP: design and implementation of a BPMN extension for clinical pathways," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM) Belfast, UK*, pp. 9–16, IEEE Computer Society, 2014.

17. E. Rolón, E. R. Aguilar, F. García, F. Ruiz, M. Piattini, L. Calahorra, M. García, and R. Martin. Process modeling of the health sector using BPMN: A case study. In *Proc. First International Conference on Health Informatics, HEALTHINF 2008, Funchal, Portugal*, vol. 2 pp. 173-178, 2008.
18. National Institute for Health and Care Excellence (NICE). Guidance and advice list. Online, 2016.
19. National Collaborating Centre for Chronic Conditions. Type 2 diabetes: national clinical guideline for management in primary and secondary care (update). Technical report, London: Royal College of Physicians, 2008.
20. National Clinical Guideline Centre. Chronic obstructive pulmonary disease: management of chronic obstructive pulmonary disease in adults in primary and secondary care. Technical report, 2010.
21. S. T. Hawley, B. Zikmund-Fisher, P. Ubel, A. Jancovic, T. Lucas, and A. Fagerlin. The impact of the format of graphical presentation on health-related knowledge and treatment choices. *Patient Education and Counseling*, 73(3):448-455, 2008.
22. Department of Health. Long Term Conditions Compendium of Information: Third Edition, Online White Paper, 2012.
23. B. Guthrie, K. Payne, P. Alderson, M. E. T. McMurdo, and S. W. Mercer. Adapting clinical guidelines to take account of multimorbidity. *BMJ: British Medical Journal*, 345(e6341), 2012.
24. National Institute for Health and Care Excellence (NICE). Multimorbidity: clinical assessment and management, NICE guideline [NG56], 2016.
25. L. D. Hughes, M. E. McMurdo, and B. Guthrie. Guidelines for people not for diseases: the challenges of applying UK clinical guidelines to people with multimorbidity. *Age and Aging*, 42(1):62-69, 2013. Oxford University Press.
26. C. Kenning, L. Fisher, P. Bee, P. Bower, and P. Coventry. Primary care practitioner and patient understanding of the concepts of multimorbidity and self-management: A qualitative study. *SAGE Open Medicine*, eCollection, 2013.
27. F. Kossak, C. Illibauer, V. Geist, J. Kubovy, C. Natschläger, T. Ziebermayr, T. Kopetzky, B. Freudenthaler, and K. Schewe. *A Rigorous Semantics for BPMN 2.0 Process Diagrams*. Springer, 2014.
28. P. Van Gorp and R. M. Dijkman, "A visual token-based formalization of BPMN 2.0 based on in-place transformations," *Information & Software Technology*, 55(2):365–394, 2013.
29. J. Ye, S. Sun, L. Wen, and W. Song, "Transformation of BPMN to YAWL," in *Computer Science and Software Engineering, (CSSE), Vol. 2: Software Engineering, Wuhan, China*, pp. 354–359, IEEE Computer Society, 2008.
30. J. Ye, S. Sun, W. Song, and L. Wen, "Formal semantics of BPMN process models using YAWL," in *International Symposium on Intelligent Information Technology Application*, 2:70–74, 2008.
31. G. Decker, R. M. Dijkman, M. Dumas, and L. García-Bañuelos, "Transforming BPMN diagrams into YAWL nets," in *Proc. Business Process Management (BPM)*, Milan, Italy, LNCS 5240:386–389, Springer, 2008.
32. P. Eklund, M. Johansson, J. Karlsson, and R. Åström, "BPMN and its semantics for information management in emergency care," in *Proc. International Conference on Computer Sciences and Convergence Information Technology*, pp. 273–278, 2009.
33. A. Awad, G. Decker, and N. Lohmann. Diagnosing and repairing data anomalies in process models. In *Business Process Management Workshops, BPM, Revised Papers, Ulm, Germany*, LNBIP 43:5-16. Springer, 2009.
34. OMG. Business Process Model and Notation (BPMN). Technical Report formal/2011-01-03, OMG, 2011.
35. Z. Wang, A. H. M. ter Hofstede, C. Ouyang, M. T. Wynn, J. Wang, and X. Zhu. How to guarantee compliance between workflows and product lifecycles? *Information Systems*, 42:195-215, 2014.
36. J. Vanhatalo, H. Völzer, and F. Leymann. Faster and more focused control-flow analysis for business process models through SESE decomposition. In *Proc. Service-Oriented Computing – ICSOC 2007, Fifth International Conference, Vienna*, LNCS 4749:43-55. Springer, 2007.
37. K. Jensen and L. M. Kristensen. *Coloured Petri Nets – Modelling and Validation of Concurrent Systems*. Springer, 2009.
38. D. Jackson. *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, London, England, 2006.
39. L. M. de Moura and N. Bjørner, "Z3: an efficient SMT solver," in *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)* Budapest, Hungary, LNCS 4963:337–340, Springer, 2008.
40. National Institute for Health and Care Excellence (NICE). Managing Stable COPD. Online, 2016.
41. National Institute for Health and Care Excellence (NICE). Management of osteoarthritis. Online, 2014.
42. J. F. Committee. *British National Formulary (BNF)*. BMJ Publishing Group Ltd and Royal Pharmaceutical Society, 72 ed., 2016.
43. M. Havey. *Essential Business Process Modeling*. O'Reilly Media, Inc., 2005.
44. W. Reisig. *Petri Nets: An Introduction*. Springer, Berlin, Germany, 1985.
45. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541-580, 1989.
46. W. M. P. van der Aalst. The application of Petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21-66, 1998.
47. R. M. Dijkman, M. Dumas, and C. Ouyang. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, 50(12):1281-1294, 2008.
48. W. M. P. van der Aalst, A. Hirnschall, and H. M. W. (Eric) Verbeek. An alternative way to analyze workflow graphs. In *Proc. Advanced Information Systems Engineering, 14th International Conference, CAiSE 2002, Toronto*, LNCS 2348:535-552, Springer, 2002.
49. F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Conceptual modelling of workflows. In *Proc. OOER'95: Object-Oriented and Entity-Relationship Modelling, 14th International Conference, Gold Coast, Australia*, LNCS 1021:341-354. Springer, 1995.
50. W. Sadiq and M. E. Orlowska. Applying graph reduction techniques for identifying structural conflicts in process models. In *Proc. Advanced Information Systems Engineering, 11th International Conference CAiSE'99, Heidelberg, Germany*, LNCS 1626:195-209. Springer, 1999.
51. M. P. Eccles, J. M. Grimshaw, P. Shekelle, H. J. Schünemann, and S. W. S. Developing clinical practice guidelines: target audiences, identifying topics for guidelines, guideline group composition and functioning and conflicts of interest. *Implementation Science*, 7(1):60, 2012.

DOI: 10.1002/smr

52. S. Woolf, H. J. Schünemann, M. P. Eccles, J. M. Grimshaw, and P. Shekelle. Developing clinical practice guidelines: types of evidence and outcomes; values and economics, synthesis, grading, and presentation and deriving recommendations. *Implementation Science*, 7(1):61, 2012.
53. P. Shekelle, S. Woolf, J. M. Grimshaw, H. J. Schünemann, and M. P. Eccles. Developing clinical practice guidelines: reviewing, reporting, and publishing guidelines; updating guidelines; and the emerging issues of enhancing guideline implementability and accounting for comorbid conditions in guideline development. *Implementation*, 7(1):62, 2012.
54. P. Gooch and A. Roudsan. Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Information Association*, 18(6):738-748, 2011.
55. R. Williams, I. E. Buchan, M. Prosperi, and J. Ainsworth. Using string metrics to identify patient journeys through care pathways. *AMIA Annual Symposium Proceedings*, pp. 1208-1217, 2014.
56. D. Demner-Fushman, W. W. Chapman, and C. J. McDonald. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5):760-772, 2009.
57. M. Taboada, M. Meizoso, D. Martínez, D. Riaño, and A. Alonso. Combining open-source natural language processing tools to parse clinical practice guidelines. *Expert Systems*, 30(1):3-11, 2013.
58. S. Meystre and P. J. Haug. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. *Journal of Biomedical Informatics*, 39(6):589-599, 2006.
59. D. S. Carrell, D. Cronkite, R. E. Palmer, K. Saunders, D. E. Gross, E. T. Masters, T. R. Hylan, and M. V. Korff. Using natural language processing to identify problem usage of prescription opioids. *International Journal of Medical Informatics*, 84(12):1057-1064, 2015.
60. E. Börger, "Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL," *Software and System Modeling*, 11(3):305–318, 2012.
61. L. Stroppi, O. Chiotti, and P. Villareal, "Extending BPMN 2.0: Method and tool support," *LNBIP*, 95:59–73, 2011.
62. W. M. P. van der Aalst and A. H. M. ter Hofstede, "YAWL: yet another workflow language," *Information Systems*, 30(4):245–275, 2005.
63. M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of Biomedical Informatics*, 46(4):744–763, 2013.
64. D. R. Sutton and J. Fox, "The syntax and semantics of the PROforma guideline modeling language," *Journal of the American Medical Informatics Association*, 10(5):433–443, 2003.
65. T. A. Pryor and G. Hripcsak, "The Arden syntax for medical logic modules," *International Journal of Clinical Monitoring and Computing*, 10(4):215–224, 1993.
66. A. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. A. Greenes, and E. H. Shortliffe, "GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines," *Journal of Biomedical Informatics*, 37(3):147–161, 2004.
67. R. Braun, H. Schlieter, M. Burwitz, and W. Esswein, "Extending a business process modeling language for domain-specific adaptation in healthcare," in *Smart Enterprise Engineering: 12. Internationale Tagung Wirtschaftsinformatik (WI), Osnabrück, Germany*, pp. 468–481, 2015.
68. J. Bowles and D. A. Meedeniya. Formal transformation from sequence diagrams to Coloured Petri Nets. In *17th Asia Pacific Software Engineering Conference, APSEC, Sydney*, pp. 216-225. IEEE Computer Society, 2010.
69. A. Meyer, S. Smirnov, and M. Weske. Data in Business Processes. TR 50, Hasso Plattner Institute at the University of Potsdam, 2011.
70. A. Meyer, L. Pufahl, D. Fahland, and M. Weske. Modeling and enacting complex data dependencies in business processes. In *Proc. Business Process Management (BPM), Beijing*, LNCS 8094:171-186. Springer, 2013.
71. Y. Sun, J. Su, B. Wu, and J. Yang. Modeling data for business processes. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE, IL, USA*, pp. 1048-1059. IEEE Computer Society, 2014.
72. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5-51, 2003.
73. C. Favre and H. Völzer. The difficulty of replacing an inclusive OR-join. In *Proc. Business Process Management – 10th International Conference, BPM, Tallinn, Estonia*, LNCS 7481:156-171. Springer, 2012.
74. M. Duerden, D. Millson, A. Avery, and S. Smart, "The quality of GP prescribing," 2011. The Kings Fund.
75. J. L. Green, J. N. Hawley, and K. J. Rask, "Is the number of prescribing physicians an independent risk factor for adverse drug events in an elderly outpatient population?," *Americal Journal of Geriatric Pharmacotherapy*, 5:31–39, 2007.
76. H. Nazar, Z. Nazar, J. Simpson, A. Yeung, and C. Whittlesea, "Use of a service evaluation and lean thinking transformation to redesign an NHS 111 refer to community pharmacy for emergency repeat medication supply service (PERMSS)," *BMJ Open*, 6(8):e011269, 2016.
77. S. Dumbreck, A. Flynn, N. Nairn, M. Wilson, S. Treweek, S. W. Mercer, P. Alderson, A. Thompson, K. Payne, and B. Guthrie, "Drug-disease and drug-drug interactions: systematic examination of recommendations in 12 UK national clinical guidelines," *British Medical Journal*, 350:h949, 2015.
78. B. A. Stewart, S. Fernandes, E. Rodriguez-Huertas, and M. Landzberg, "A preliminary look at duplicate testing associated with lack of electronic health record interoperability for transferred patients," *Journal of the American Medical Informatics Association*, 17(3):314-344, 2010.
79. I. Bardhan, S. Ayabakan, E. Zheng, and K. Kirksey, "Value of health information sharing in reducing healthcare waste: An analysis of duplicate testing across hospitals," in *International Conference on Information Systems (ICIS)*, 2014.
80. S. Davies, A. Umranikar, T. Huggins, A. Gauthier, and G. T. Harty, "Cost implications of adapting the investigation and diagnosis pathway of infertilty patients in a UK NHS setting," *Value in Health*, 19(7):A625, 2016.
81. A. Burattin. PLG2: Multiperspective processes randomization and simulation for online and offline settings. *CoRR*, abs/1506.08415, 2015.

82. C. M. Boyd, J. Darer, C. Boult, L. P. Fried, L. Boult, and A. W. Wu, "Clinical practice guidelines and quality of care for older patients with multiple comorbid diseases: Implications for pay for performance," *Journal of the American Medical Association*, 294(6):716–724, 2005.
83. C. Salisbury, L. Johnson, S. Purdy, J. M. Valderas, and A. A. Montgomery, "Epidemiology and impact of multimorbidity in primary care: a retrospective cohort study," *British Journal of General Practice*, 61(582):e12–21, 2011.
84. K. Barnett, S. W. Mercer, M. Norbury, G. Watt, S. Wyke, and B. Guthrie, "Epidemiology of multimorbidity and implications for health care, research, and medical education: a cross-sectional study," *The Lancet*, 380(9836):37–43, 2012.
85. A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen, and K. Jensen. CPN tools for editing, simulating, and analysing Coloured Petri Nets. In *Proc. Applications and Theory of Petri Nets ICATPN, Eindhoven, The Netherlands*, LNCS 2679:450-462. Springer, 2003.
86. K. Jensen, L. M. Kristensen, and L. Wells. Coloured Petri Nets and CPN tools for modelling and validation of concurrent systems. *STTT*, 9(3-4):213-254, 2007.
87. M. Westergaard, S. Evangelista, and L. M. Kristensen. ASAP: an extensible platform for state space analysis. In *Proc. Applications and Theory of Petri Nets, Paris*, LNCS 5606:303-312. Springer, 2009.
88. M. Alwanain, B. Bordbar, and J. K. F. Bowles. Automated composition of sequence diagrams via Alloy. In *Proc. MODELSWARD – Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development, Lisbon*, pp. 384-391. SciTePress, 2014.

DOI: 10.1002/smr